M 7

```
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB      RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB      RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB      RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
CCC              000         000  BBB         BBB   RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB   RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB   RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB   RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB   RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB   RRR         RRR        TTT        LLL
CCC              000         000  BBBBBBBBBBBB      RRRRRRRRRRRR           TTT        LLL
CCC              000         000  BBBBBBBBBBBB      RRRRRRRRRRRR           TTT        LLL
CCC              000         000  BBBBBBBBBBBB      RRRRRRRRRRRR           TTT        LLL
CCC              000         000  BBB         BBB   RRR   RRR              TTT        LLL
CCC              000         000  BBB         BBB   RRR   RRR              TTT        LLL
CCC              000         000  BBB         BBB   RRR   RRR              TTT        LLL
CCC              000         000  BBB         BBB   RRR      RRR           TTT        LLL
CCC              000         000  BBB         BBB   RRR      RRR           TTT        LLL
CCC              000         000  BBB         BBB   RRR      RRR           TTT        LLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB      RRR         RRR        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB      RRR         RRR        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB      RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
                                        E  1
**FILE**ID**COBHANDLE
```

```
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH    AAAAAA    NN      NN  DDDDDDDD    LL          EEEEEEEEEE
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH    AAAAAA    NN      NN  DDDDDDDD    LL          EEEEEEEEEE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NNNN    NN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NNNN    NN  DD      DD  LL          EE
CC          OO      OO  BBBBBBBB    HHHHHHHHHH  AA      AA  NN  NN  NN  DD      DD  LL          EEEEEEEE
CC          OO      OO  BBBBBBBB    HHHHHHHHHH  AA      AA  NN  NN  NN  DD      DD  LL          EEEEEEEE
CC          OO      OO  BB      BB  HH      HH  AAAAAAAAAA  NN    NNNN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AAAAAAAAAA  NN    NNNN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE               ....
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE               ....
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH  AA      AA  NN      NN  DDDDDDDD    LLLLLLLLLL  EEEEEEEEEE       ....
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH  AA      AA  NN      NN  DDDDDDDD    LLLLLLLLLL  EEEEEEEEEE       ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE COB$$HANDLER (
0                 IDENT = '1-022'          ! FILE: COBHANDLE.B32 EDIT:PDG1022
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !
1 !++
1 ! FACILITY:      COBOL SUPPORT
1 !
1 ! ABSTRACT:      This procedure is the error handler for COBOL error
1 !                conditions.  It gets invoked as a result of a call
1 !                to LIB$SIGNAL.
1 !
1 !
1 ! ENVIRONMENT:  Vax-11 User Mode
1 !
1 ! AUTHOR: MLJ , CREATION DATE: 03-MAY-1979
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original. MLJ  03-MAY-1979
1 ! 1-002 - Added boilerplate and comments. RKR 18-JULY-1979
1 ! 1-003 - Declare psects via library macro. RKR 23-AUG-1979
1 ! 1-004 - Change symbolic name of LIBRARY file. RKR 1-OCT-79
1 ! 1-005 - Change name of entry point to COB$$HANDLER. RKR 18-OCT-79
1 ! 1-006 - Remove definition of COB$_USE_EXIT. Cosmetic changes.
1 !         RKR 20-OCT-79
1 ! 1-007 - But forgot to declare it as EXTERNAL LITERAL.
1 !         RKR 20-OCT-79
1 ! 1-008 - Check for COB$_USE_EXIT by using LIB$MATCH_COND.
1 !         RKR 22-OCT-79
1 ! 1-009 - Make arguments to LIBMATCHCOND be by REF.  RKR 31-OCT-79
1 ! 1-010 - Add code to resignal a COBOL-specific error message if
1 !         signal is SS$_ROPRAND and associated opcode was
```

```
1 !             CVTTP or CVTSP.  RKR 21-NOV-79
1 ! 1-011 - Correct resignaling code.  RKR 27-NOV-79
1 ! 1-012 - Added abstract, functional description, comments and made
1 !             cosmetic changes.  Added code that breaks up the CASE code
1 !             for a USE procedure condition value into the appropriate types -
1 !             file specific and mode specific.  This addition of code
1 !             also involved introducing two new condition values and
1 !             symbolic names for those values.  LB  3-MAR-81
1 ! 1-013 - Added comments.  LB 09-MAR-81
1 ! 1-014 - Added code for handling a data base USE procedure condition
1 !             code (as a result of the new routine COB$DBEXCEPTION).  Also
1 !             changed the range of the CASE statement from 0-3 to 0-5 to
1 !             account for new error conditions.  And added yet more
1 !             comments.  LB  12-MAR-81
1 ! 1-015 - Added code to check for the DB code (check that COB$B_USE_CODE
1 !             equals COB$K_DBUSE_CODE) before searching for the DB entry.
1 !             This code used to reside in routine COB$DBEXCEPTION.  LB  16-MAR-81
1 ! 1-016 - Replaced arbitrary signalled values for USE procedure checking
1 !             code with appropriate symbol names which are now defined in
1 !             COBMSGDEF.  Added corresponding entries in the EXTERNAL LITERAL
1 !             declarations for this module.  LB  24-MAR-81
1 ! 1-017 - Changed names of the external literals to correspond to changes
1 !             made in COBMSG.MDL.  Deleted call to LIB$MATCH_COND and changed
1 !             the CASE stmt to a SELECTONE stmt.  Changed labels in the
1 !             SELECTONE stmt (used to be a CASE) to be mnemonics instead of
1 !             numbers.  Added comments.  LB  16-APR-81
1 ! 1-018 - Deleted the external literals COB$_LSTHNDLOP and LSTHNDLFL and
1 !             added LSTHNDUSE.  This was done as a result of a change made
1 !             in COBOL regarding the scoping rules for USE procedures.  Also
1 !             changed the macro name for the signalling arguments in the signal
1 !             array to reflect changes made to COBDEF (the reference had been
1 !             [COB$A_CHK_PROC] which has been changed and extended to the fields
1 !             [COB$A_OPN_PROC] and [COB$A_FIL_PROC]).  LB  21-APR-81
1 ! 1-019 - Entry point changed to COB$$HANDLER.  For some reason, it had
1 !             remained a single $ entry point.  Resolves duplicate symbol
1 !             problem with COBDHANDL.  LB 3-AUG-81
1 ! 1-020 - Added external routine declaration for COB$HANDLER.  LB 4-AUG-81
1 !
1 ! 1-021 - Added handling of SORT/MERGE signalled errors.  Currently
1 !             using literal SORT_FAC_CODE until the literal SORT$_FACILITY
1 !             is put into STARLET.  ER 16-MAR-84
1 ! 1-022 - Move handling of SORT/MERGE errors to end of SELECTONE, and
1 !             resignal the errors prefixed with the COB$_ERRDURSOR message.
1 !             Remove unreferenced variables.  Change indentation, and reword the
1 !             checks that validate the addresses of the USE lists.  Add comments.
1 !             PDG 9-Apr-84
1 !--
1
1 !<BLF/PAGE>
```

```
1 !+
1 ! SWITCHES
1 !-
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !+
1 ! LINKAGES
1 !       NONE
1 !-
1
1 !+
1 ! TABLE OF CONTENTS:
1 !-
1
1 FORWARD ROUTINE
1
1 ! changed name to cob$$handler
1         COB$$HANDLER;
1
1 !+
1 ! INCLUDE FILES
1 !-
1
1 REQUIRE 'RTLIN:RTLPSECT';                       ! Macros for declaring psects
1 LIBRARY 'RTLSTARLE';                            ! RTL routines
1 REQUIRE 'RTLIN:COBDEF';                         ! COBOL specific RTL macros and literals
1
1 !+
1 ! MACROS
1 !       NONE
1 !-
1
1 !+
1 ! EQUATED SYMBOLS
1 !-
1
1 LITERAL
1         CVTTP_OPCODE = %X'26',                  ! Opcode value for CVTTP instruction
1         CVTSP_OPCODE = %X'09';                  ! Opcode value for CVTSP instruction
1
1 !+
1 ! PSECT DECLARATIONS:
1 !-
1
1 DECLARE_PSECTS (COB);                           ! Psects for COB$ facility
1
1 !+
1 ! EXTERNAL REFERENCES
1 !-
1
1 EXTERNAL ROUTINE
1         LIB$STOP,
1         LIB$SIGNAL,
1         COB$$INVOKE_USE: NOVALUE,               ! Invoke the USE procedure
1         COB$HANDLER;
1
```

```
1 EXTERNAL LITERAL
1               COB$_ERRDURSOR,
1               COB$_NO_USEPRO,                 ! No USE procedure available - error on file !AS
1               COB$_LSTHNDUSE,                 ! Lost handler for a USE procedure - environment corrupted !2(+)
1               COB$_LSTHNDLDB,                 ! Lost handler for data base exception - environment corrupted !+
1               COB$_USE_EXIT,                  ! Special signal from COB$$INVOKE_USE
1               COB$_INVDECDAT;                 ! Invalid decimal data signal
1 LITERAL
1               SORT$_FACILITY = 28;            ! Delete when SORT$_FACILITY is in STARLET 1-021 ER
```

```
1 ! changed name to cob$$handler
1 GLOBAL ROUTINE COB$$HANDLER(SIGNAL,MECHANISM)=
1
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is the error handler for COBOL error conditions.
1 !       It gets invoked as a result of a call to LIB$SIGNAL.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       COB$$HANDLER (signal.rr.r, mechanism.rr.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIGNAL.rr.r     Address of vector of longwords indicating
1 !                       nature of condition.
1 !
1 !       MECHANISM.rr.r  Address of vector of longwords indicating
1 !                       the state of the process.
1 !
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 ! NOTES:
1 !
1 !       The macro field references beginning with 'CHF' refer to
1 !       condition handling argument list offsets.
1 !
1 !--
```

```
    BEGIN

    MAP
        MECHANISM:      REF BLOCK[,BYTE],
        SIGNAL:         REF BLOCK[,BYTE];
    LOCAL
        CONDITION;      ! Condition that was signalled
    LITERAL
        FALSE = 0,
        TRUE = 1;
!+
! Find out if it is a signal of interest.
!-


    CONDITION = .SIGNAL[CHF$L_SIG_NAME];        ! Fetch condition value from signal array


!+
! Select appropriate action based on which one we have.
!-

    SELECTONE .CONDITION OF
        SET

        [COB$_USE_EXIT]:        ! Special COBOL unwind signal

            $UNWIND();          ! Just unwind

        [SS$_ROPRAND]:          ! Was a SS$_ROPRAND
            BEGIN

            LOCAL
                USER_PC;        ! Program counter where exception took place

                !+
                ! Using the signal argument vector, extract the
                ! program counter at the time the SS$_ROPRAND occurred.
                ! The PC is the second-to-last argument in the signal vector.
                ! Note that %BPVAL and %UPVAL are pre-declared BLISS literals
                ! defining bits per BLISS value and units per BLISS value.
                !-

            USER_PC = .SIGNAL[ (.SIGNAL[CHF$L_SIG_ARGS]-1)*%UPVAL,0,%BPVAL,0 ];

                !+
                ! Check to see if a CVTTP or a CVTSP instruction
                ! was the generator of the signal.  Note that the
                ! PC is pointing to the instruction that caused the fault.
                ! If the debugger had a breakpoint set on this instruction,
                ! this check will fail, but that's okay.
                !-

            IF (   .(.USER_PC)<0,8> EQL CVTTP_OPCODE  OR
                   .(.USER_PC)<0,8> EQL CVTSP_OPCODE      )
            THEN
        !+
        ! Set the first longword of the signal argument
        ! vector (the condition value field) to the condition
        ! name we want the user to see, invalid decimal data.
        !-

        SIGNAL[CHF$L_SIG_NAME] = COB$_INVDECDAT;
    END;

!+
! The following code handles the case of file specific
! and open-mode specific GLOBAL USE procedure conditions.
! Search through the entries for a match.  If there is a match,
! then invoke the USE procedure and return SS$_CONTINUE;
! otherwise, re-signal the error (return SS$_RESIGNAL).
!-

[COB$_LSTHNDUSE]:
    BEGIN

    LOCAL
        FP:             REF BLOCK[,BYTE],
        SFP:            REF BLOCK[,BYTE];         ! Saved FP
    REGISTER
        USE = 2:        REF BLOCK[,BYTE],         ! Pointer to USE list
        USEENT = 3:     REF BLOCK[,BYTE];         ! Pointer to USE list entry

    FP = .MECHANISM[CHF$L_MCH_FRAME];             ! Get FP of this program
    SFP = .FP[SF$L_SAVE_FP];                      ! Get FP of program we want to look at
    IF  BEGIN
        !+
        ! This check is to ensure that the only way
        ! you could get here is from a COBOL program.
        ! Note that we check for COB$HANDLER, rather than COB$$HANDLER;
        ! COB$HANDLER is the symbol that COBOL programs reference - it
        ! may be in a transfer vector or a fixup section; ie, the frame
        ! may not hold a direct reference to COB$$HANDLER.
        ! Also, get the USE list.
        ! Note that, if we get here, the USE list won't be zero.
        !-
        IF .SFP EQL 0 THEN FALSE
        ELSE IF .SFP[SF$A_HANDLER] NEQA COB$HANDLER THEN FALSE
        ELSE IF (USE = .SFP[COB$A_SF_USE]) EQL 0 THEN FALSE
        ELSE TRUE
        END
    THEN
        BEGIN

        !+
        ! Search for a USE procedure declared for the specific file
        ! on which the exception occurred.  Note that the
        ! COB$A_USE_FILES reference is the base of the 1st file
        ! entry and COB$B_GUSE_COUNT is the count of global
        ! file entries.
        !-

        USEENT = USE[COB$A_USE_FILES];  ! Point to first file entry
```

```
4          DECR I FROM .USE[COB$B_GUSE_COUNT]-1 TO 0 DO
5              BEGIN                                   ! Loop over files
5              IF .USEENT[COB$A_USE_PROC] EQLA .SIGNAL[COB$A_FIL_PROC]
5              THEN
6                  BEGIN
6                  COB$$INVOKE_USE(                    ! Invoke USE
6                      .USEENT[COB$A_USE_PROC],        ! Addr of USE procedure
6                      .USE,                           ! Ptr to USE list
6                      .FP[SF$L_SAVE_AP],              ! Argument pointer
6                      .USEENT[COB$A_USE_EOPR],        ! Addr of EOPR block
6                      .USE[COB$A_USE_PN[]);           ! Addr of Perform Nest Ctr
6                  RETURN SS$_CONTINUE;
5                  END;
5              USEENT = .USEENT + COB$S_USE_FILES; ! Step to next
4              END;
4
4
4          !+
4          ! Open Mode Only.
4          !
4          ! See if a procedure has been declared for the
4          ! open mode.  Note that COB$A_USE_MODES refers to
4          ! the base of the open mode entries.  There are four
4          ! open modes, i.e. INPUT, OUTPUT, I-O, and EXTEND.
4          !-
4
4          USEENT = USE[COB$A_USE_MODES];  ! Point to first mode entry
4          DECR I FROM 3 TO 0 DO
5              BEGIN                                   ! Loop over modes
5
5              !+
5              ! The check here for EOPR not equal to zero is
5              ! to ensure that the program is a local one.  If
5              ! EOPR equals zero, then the USE procedure is an
5              ! up-level reference; then the original condition
5              ! that was signalled, should be re-signalled.
5              ! Else, if EOPR is not equal to zero and the USE
5              ! procedure has been found, then call COB$INV_USE.
5              !-
5
5              IF .USEENT[COB$A_USE_PROC] EQLA .SIGNAL[COB$A_OPN_PROC]
5                  AND .USEENT[COB$A_USE_EOPR] NEQ 0
5              THEN
6                  BEGIN
6                  COB$$INVOKE_USE(                    ! Invoke USE
6                      .USEENT[COB$A_USE_PROC],        ! Addr of USE procedure
6                      .USE,                           ! Ptr to USE list
6                      .FP[SF$L_SAVE_AP],              ! Argument ptr
6                      .USEENT[COB$A_USE_EOPR],        ! Addr of EOPR block
6                      .USE[COB$A_USE_PN[]);           ! Addr of Perform Nest Ctr
6                  RETURN SS$_CONTINUE;
5                  END;
5              USEENT = .USEENT + COB$S_USE_MODES; ! Step to next
4              END;
3          END;
2      END;
2
```

```
            !+
            ! The following code handles the case of a Data Base
            ! GLOBAL USE procedure condition.  Search through the COB$GDBUSE_CNT
            ! entries and check for a match for the entry address with the
            ! address of the USE procedure passed to this handler in the
            ! signal argument vector.  If there is a match, then invoke the
            ! USE procedure and return SS$_CONTINUE; otherwise, re-signal the
            ! error (return SS$_RESIGNAL).
            !-

        [COB$_LSTHNDLDB]:
            BEGIN

            LOCAL
                FP:             REF BLOCK[,BYTE],
                SFP:            REF BLOCK[,BYTE];       ! Saved FP
            REGISTER
                USE = 2:        REF BLOCK[,BYTE],       ! Ptr to Data Base USE list
                USEENT = 3:     REF BLOCK[,BYTE];       ! Ptr to Data Base USE list entry

            FP = .MECHANISM[CHF$L_MCH_FRAME];           ! Get FP of this program
            SFP = .FP[SF$L_SAVE_FP];                    ! Get FP of program we want to look at
            IF  BEGIN
                !+
                ! This check is to ensure that the only way
                ! you could get here is from a COBOL program.
                ! Also, get the DB USE list.
                !
                ! Check if this is a DB USE list.
                ! The COB$B_USE_CODE field should contain
                ! the generic code for the class of data base
                ! exceptions (equal to COB$K_DBUSE_CODE).
                ! This allows new kinds of USE procedures to be added,
                ! without requiring more longwords on the COBOL stack frame.
                !-
                IF .SFP EQL 0 THEN FALSE
                ELSE IF .SFP[SF$A_HANDLER] NEQA COB$HANDLER THEN FALSE
                ELSE IF (USE = .SFP[COB$A_DB_USE]) EQL 0 THEN FALSE
                ELSE IF .USE[COB$B_USE_CODE] NEQ COB$K_DBUSE_CODE THEN FALSE
                ELSE TRUE
                END
            THEN
                BEGIN

                !+
                ! Search for a USE procedure for the corresponding
                ! Data Base exception.  Note that the COB$A_DB_USE
                ! reference is the address of the data base entry while
                ! COB$B_DBUSE_CNT is the count of global Data Base
                ! USE procedures defined in the local program.
                !-

                USEENT = USE[COB$A_DBUSE_ENT];  ! Point to 1st data base entry
                DECR I FROM .USE[COB$B_GDBUSE_CNT] - 1 TO 0 DO
                    BEGIN
                    IF .USEENT[COB$A_USE_PROC] EQLA .SIGNAL[COB$A_DBCHK_PROC]
                    THEN
```

```
6                               BEGIN
6                               COB$$INVOKE_USE (               ! Invoke the USE procedure
6                                   .USEENT[COB$A_USE_PROC],    ! Addr of DB USE procedure
6                                   .USE,                       ! Ptr to DB USE list
6                                   .FP[SF$L_SAVE_AP],          ! Argument pointer
6                                   .USEENT[COB$A_USE_EOPR],    ! Addr of EOPR block
6                                   .USE[COB$A_DBUSE_PNC]);     ! Addr of Perform Nest Ctr
6                               RETURN SS$_CONTINUE;
5                               END;
5                           USEENT = .USEENT + COB$S_DBUSE;     ! Step to next entry
4                           END;
3                       END;
2                   END;
2
2               !+
2               ! Check for other errors that are handled specially.
2               ! Currently, these only include errors from Sort/Merge.
2               !-
2
2               [OTHERWISE]:                            ! No match occurred
3                   BEGIN
3                   MAP
3                       CONDITION:      BLOCK[,BYTE];   ! Condition that was signalled
3
3                   !+
3                   ! Is it a SORT/MERGE error signal?
3                   !-
3                   IF  .CONDITION[STS$V_FAC_NO] EQL SORT$_FACILITY
3                   THEN
4                       BEGIN
4                       IF .CONDITION[STS$V_SEVERITY] LSS STS$K_SEVERE
4                       THEN
5                           BEGIN
5                           !
5                           ! These errors are continuable.
5                           !
5                           RETURN SS$_CONTINUE         ! Ignore the error
5                           END
4                       ELSE
5                           BEGIN
5                           !
5                           ! Resignal the error, prefixing the Cobol-specific
5                           ! error message, and removing the PC and PSL.
5                           !
5                           ! Note that, although we don't need to increase the size
5                           ! of the SIGNAL vector, we can't use it for the new signal,
5                           ! since we musn't just mung the PC and PSL in this vector.
5                           !
5                           ! We assume that ARG_K_SIZE longwords suffice.  If not,
5                           ! the displayed message will look tacky, that's all.
5                           !
5                           LITERAL ARG_K_SIZE = 12;            ! Should be large enough
5                           LOCAL ARGS: VECTOR[ARG_K_SIZE];
5                           BUILTIN CALLG;
5
5                           ARGS[0] = MINU(.SIGNAL[CHF$L_SIG_ARGS], ARG_K_SIZE-1);
5                           ARGS[1] = COB$_ERRDURSOR;
```

```
5                              ARGS[2] = 0;

5                              CH$MOVE(.ARGS[0] * %UPVAL   ! Everything ...
5                                      - 2*%UPVAL,          ! Less bytes for PC and PSL
5                                  SIGNAL[CHF$L_SIG_NAME],
5                                  ARGS[3]);
5                              CALLG( ARGS[0], LIB$STOP );
5
5                              RETURN SS$_CONTINUE          ! Ignore the original error
5
4                              END;
3                          END;
2                      END;
2
2                  TES;
2
2         !+
2         ! Resignal the error if the signalled condition was not one
2         ! of the expected conditions to be handled.  Also resignal the
2         ! error if a USE procedure wasn't found or if the error had been
2         ! a SS$_ROPRAND since the signal name has been changed.
2         !-
2
2         RETURN SS$_RESIGNAL
1         END;


                                                        .TITLE  COB$$HANDLER
                                                        .IDENT  \1-022\

                                                        .EXTRN  LIB$STOP, LIB$SIGNAL
                                                        .EXTRN  COB$$INVOKE_USE
                                                        .EXTRN  COB$HANDLER, COB$_ERRDURSOR
                                                        .EXTRN  COB$_NO_USEPRO, COB$_LSTHNDUSE
                                                        .EXTRN  COB$_LSTHNDLDB, COB$_USE_EXIT
                                                        .EXTRN  COB$_INVDECDAT, SYS$UNWIND

                                                        .PSECT  _COB$CODE,NOWRT,  SHR,  PIC,2

                                      00FC 00000        .ENTRY  COB$$HANDLER, Save R2,R3,R4,R5,R6,R7      ; 0709
                 57 00000000G   00      9E 00002        MOVAB   COB$HANDLER, R7                          ;
                 5E             30      C2 00009        SUBL2   #48, SP                                  ;
                 54       04    AC      D0 0000C        MOVL    SIGNAL, R4                               ; 0773
                 52       04    A4      D0 00010        MOVL    4(R4), CONDITION                         ;
      00000000G  8F             52      D1 00014        CMPL    CONDITION, #COB$_USE_EXIT                ; 0783
                                0B      12 0001B        BNEQ    1$                                       ;
                                7E      7C 0001D        CLRQ    -(SP)                                    ; 0785
      00000000G  00             02      FB 0001F        CALLS   #2, SYS$UNWIND                           ;
                                7C      11 00026        BRB     9$                                       ;
      00000454   8F             52      D1 00028  1$:   CMPL    CONDITION, #1108                         ; 0787
                                1C      12 0002F        BNEQ    3$                                       ;
                 50             64      D0 00031        MOVL    (R4), R0                                 ; 0801
                 50       FC A440       D0 00034        MOVL    -4(R4)[R0], USER_PC                      ;
                 26             60      91 00039        CMPB    (USER_PC), #38                           ; 0811
                                05      13 0003C        BEQL    2$                                       ;
                 09             60      91 0003E        CMPB    (USER_PC), #9                            ; 0812
                                61      12 00041        BNEQ    9$                                       ;
```

```
             04   A4 00000000G  8F  D0 00043 2$:    MOVL    #COB$_INVDECDAT, 4(R4)                ; 0821
                                57  11 0004B        BRB     9$                                    ; 0780
      00000000G   8F            52  D1 0004D 3$:    CMPL    CONDITION, #COB$_LSTHNDUSE            ; 0832
                                50  12 00054        BNEQ    10$                                   ;
                  50        08  AC  D0 00056        MOVL    MECHANISM, R0                         ; 0842
                  55        04  A0  D0 0005A        MOVL    4(R0), FP                             ;
                  50        0C  A5  D0 0005E        MOVL    12(FP), SFP                           ; 0843
                                65  13 00062        BEQL    11$                                   ; 0855
                  51            67  9E 00064        MOVAB   COB$HANDLER, R1                       ; 0856
                  51            60  D1 00067        CMPL    (SFP), R1                             ;
                                62  12 0006A        BNEQ    12$                                   ;
                  52        FC  A0  D0 0006C        MOVL    -4(SFP), USE                          ; 0857
                                57  13 00070        BEQL    11$                                   ;
                  53        28  A2  9E 00072        MOVAB   40(R2), USEENT                        ; 0871
                  56        25  A2  9A 00076        MOVZBL  37(USE), I                            ; 0874
                                09  11 0007A        BRB     5$                                    ;
             0C   A4            63  D1 0007C 4$:    CMPL    (USEENT), 12(R4)                      ;
                                18  13 00080        BEQL    7$                                    ;
                  53            0C  C0 00082        ADDL2   #12, USEENT                           ; 0885
                  F4            56  F4 00085 5$:    SOBGEQ  I, 4$                                 ; 0872
                  53        04  A2  9E 00088        MOVAB   4(R2), USEENT                         ; 0898
                  56            03  D0 0008C        MOVL    #3, I                                 ; 0912
             10   A4            63  D1 0008F 6$:    CMPL    (USEENT), 16(R4)                      ;
                                09  12 00093        BNEQ    8$                                    ;
                            04  A3  D5 00095        TSTL    4(USEENT)                             ; 0913
                                04  13 00098        BEQL    8$                                    ;
                                62  DD 0009A 7$:    PUSHL   (USE)                                 ; 0921
                                45  11 0009C        BRB     14$                                   ; 0920
                  53            08  C0 0009E 8$:    ADDL2   #8, USEENT                            ; 0924
                  EB            56  F4 000A1        SOBGEQ  I, 6$                                 ; 0899
                                56  11 000A4 9$:    BRB     17$                                   ; 0780
      00000000G   8F            52  D1 000A6 10$:   CMPL    CONDITION, #COB$_LSTHNDLDB            ; 0939
                                4F  12 000AD        BNEQ    18$                                   ;
                  50        08  AC  D0 000AF        MOVL    MECHANISM, R0                         ; 0949
                  55        04  A0  D0 000B3        MOVL    4(R0), FP                             ;
                  50        0C  A5  D0 000B7        MOVL    12(FP), SFP                           ; 0950
                                3F  13 000BB        BEQL    17$                                   ; 0964
                  51            67  9E 000BD        MOVAB   COB$HANDLER, R1                       ; 0965
                  51            60  D1 000C0        CMPL    (SFP), R1                             ;
                                7A  12 000C3        BNEQ    21$                                   ;
                  52        F8  A0  D0 000C5        MOVL    -8(SFP), USE                          ; 0966
                                74  13 000C9 11$:   BEQL    21$                                   ;
                  01            62  91 000CB        CMPB    (USE), #1                             ; 0967
                                6F  12 000CE 12$:   BNEQ    21$                                   ;
                  53        0C  A2  9E 000D0        MOVAB   12(R2), USEENT                        ; 0981
                  56        09  A2  9A 000D4        MOVZBL  9(USE), I                             ; 0984
                                1F  11 000D8        BRB     16$                                   ;
             0C   A4            63  D1 000DA 13$:   CMPL    (USEENT), 12(R4)                      ;
                                16  12 000DE        BNEQ    15$                                   ;
                            04  A2  DD 000E0        PUSHL   4(USE)                                ; 0992
                            04  A3  DD 000E3 14$:   PUSHL   4(USEENT)                             ; 0991
                            08  A5  DD 000E6        PUSHL   8(FP)                                 ; 0990
                                52  DD 000E9        PUSHL   USE                                   ; 0989
                                63  DD 000EB        PUSHL   (USEENT)                              ; 0988
      00000000G   00            05  FB 000ED        CALLS   #5, COB$$INVOKE_USE                   ;
                                45  11 000F4        BRB     20$                                   ; 0993
                  53            0C  C0 000F6 15$:   ADDL2   #12, USEENT                           ; 0995
```

```
                                  DE         56  F4 000F9 16$:   SOBGEQ  I, 13$                                      ; 0982
                                             41  11 000FC 17$:   BRB     21$                                         ; 0780
        1C          52            0C         10  ED 000FE 18$:   CMPZV   #16, #12, CONDITION, #28                    ; 1013
                                             3A  12 00103        BNEQ    21$                                         ;
        04          52            03         00  ED 00105        CMPZV   #0, #3, CONDITION, #4                       ; 1016
                                             2F  19 0010A        BLSS    20$                                         ;
                                  50         64  D0 0010C        MOVL    (R4), R0                                    ; 1041
                                  0B         50  D1 0010F        CMPL    R0, #11                                     ;
                                             03  1B 00112        BLEQU   19$                                         ;
                                  50         0B  D0 00114        MOVL    #11, R0                                     ;
                                  6E         50  D0 00117 19$:   MOVL    R0, ARGS                                    ;
                         04       AE 00000000G  8F  D0 0011A     MOVL    #COB$_ERRDURSOR, ARGS+4                     ; 1042
                                        08   AE  D4 00122        CLRL    ARGS+8                                      ; 1043
                                  50         6E  D0 00125        MOVL    ARGS, R0                                    ; 1045
                                  50         04  C4 00128        MULL2   #4, R0                                      ; 1046
                                  50         08  C2 0012B        SUBL2   #8, R0                                      ;
                    0C     AE      04  A4    50  28 0012E        MOVC3   R0, 4(R4), ARGS+12                          ; 1048
                          00000000G   00     6E  FA 00134        CALLG   ARGS, LIB$STOP                              ; 1049
                                  50         01  D0 0013B 20$:   MOVL    #1, R0                                      ; 1051
                                             04 0013E            RET                                                 ; 1025
                                  50    0918  8F  3C 0013F 21$:  MOVZWL  #2328, R0                                   ; 1066
                                             04 00144            RET                                                 ; 1067

; Routine Size:  325 bytes,    Routine Base:  _COB$CODE + 0000


;   536            1068  1
;   537            1069  1 END
;   538            1070  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                     Bytes                         Attributes
;
;   _COB$CODE                          325  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                              Library Statistics
;
;                                             -------- Symbols --------       Pages         Processing
;       File                                  Total    Loaded   Percent       Mapped        Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1          9776       14         0          581           00:00.7
;                                          COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:COBHANDLE/OBJ=OBJ$:COBHANDLE MSRC$:COBHANDLE/UPDATE=(ENH$:COBHANDLE
;       )

; Size:          325 code + 0 data bytes
; Run Time:         00:08.3
; Elapsed Time:     00:36.4
; Lines/CPU Min:     7772
; Lexemes/CPU-Min: 28895
; Memory Used:  155 pages
; Compilation Complete
```

0063 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COBPAUSE
LIS

COBMSG
LIS

COBHANDLE
LIS

COBINTARI
LIS

COBINTER
LIS

COBMULQ
LIS

COBPOSERA
LIS

COBIOEXCE
LIS

COBLINAGE
LIS

COBKEY
LIS

COBINUUSE
LIS